# Time Travel Conquest

**time limit per test:** 5 seconds
**memory limit per test:** 1 gigabyte
**input:** standard input
**output:** standard output

points: 100 · problem code: ttc · author: glipR

Toki is a time traveller who lives through the various ages of mankind and witnesses the rise and fall of kingdoms. Toki can time travel only by warping herself back in time, which then starts a new branch in the timeline.

Every **city** is defended by an **army**.

Whenever two cities fight, the defender of all cities previously defended by the losing army is now defended by the winning army.

So for example, we may have:

- City A defended by Army A
- City B defended by Army B
- City C defended by Army C
- City D defended by Army B

Then if City D and City A fight, we have Army A versus Army B - Suppose Army A wins. And so now we have:

- City A defended by Army A
- City B defended by Army A
- City C defended by Army C
- City D defended by Army A

## Input

Input will consist of a variety of queries/events that occur throughout Toki's life. There are three types of queries:

- Attack queries - These represent two cities attacking. The query contains 4 values: `ATTACK`, Name of the winning city, Name of the losing city, and the year that this attack occurred.
- Reset queries - These represent Toki resetting the timeline to a particular year. The query contains 2 values: `RESET` and the year that we reset back to. (This will not reset attacks that occured on the same year)
- Defending queries - These are a question that require program output. The query simply asks what is the defender of some city in the present time. The query contains two values: `DEFENDER` and the city name.

Input will start with two space-separated integers, $C$ and $Q$. The number of cities and queries respectively.

$C$ lines of input follow, each container two strings of upper/lowercase a-z letters. These represent the name of the city, and the name of the army defending the city. $Q$ lines of input follow, each containing one of the four query types above.

## Output

Output a single integer on a new line for each of the `DEFENDER` queries provided in the input.

## Constraints

- All ATTACK statements will occur between two armies that still have cities to defend.
- $1 \leq C \leq 2*10^5$
- $1 \leq Q \leq 2*10^5$
- All years $y$ satisfy $1 \leq y \leq 10^9$
- Unless a reset occurs in between, all consecutive ATTACK statements will not decrease in years. (IE the statements should make chronological sense)

**Test Set A** (66.6%)

- The total number of `RESET` queries does not exceed 5.

**Test Set B** (33.3%)

No extra constraints

## Example Input/Output

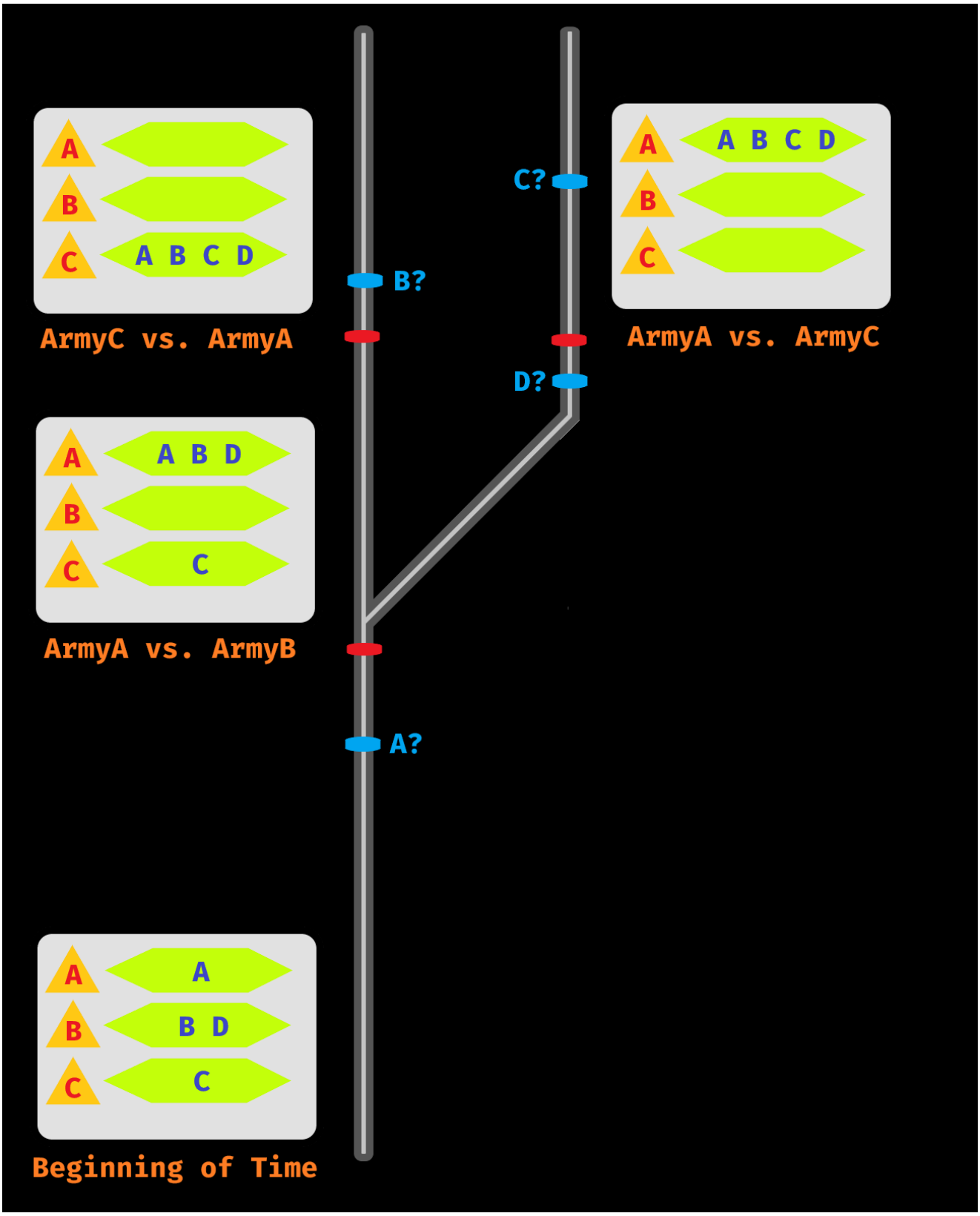
**Input**

```
4 8
CityA ArmyA
CityB ArmyB
CityC ArmyC
CityD ArmyB
DEFENDER CityA
ATTACK ArmyA ArmyB 1010
ATTACK ArmyC ArmyA 1015
DEFENDER CityB
RESET 1010
DEFENDER CityD
ATTACK ArmyA ArmyC 1015
DEFENDER CityC
```

**Output**

```
ArmyA
ArmyC
ArmyA
ArmyA
```

See the following image for epxlanation:


A
B
C
A B C D
ArmyC vs. ArmyA
B?
C?
A B C D
ArmyA vs. ArmyC
D?
A
B
C
A B D
C
ArmyA vs. ArmyB
A?
A
B
C
A
B D
C
Beginning of Time